

User Manual

Connection to CAN

Part Number: 80 860.634
Version: 2
Date: 23.11.2005
Valid for: TSwin .net 4.0x
TSwin .net 4.1x

Version	Date	Modifications
1	04.07.2005	First edition
2	23.11.2005	Validation extended, chapter "Important Notes" added, protocol parameters corrected

This manual, including all illustrations contained herein, is copyright protected. Use of this manual by any third party in departure from the copyright provision is forbidden. No part of this manual may be reproduced, translated or electronically or photographically archived or altered without the express written consent from Sutron electronic GmbH. Violations shall be cause for damage liability.
Sutron electronic reserves the right to make any changes that contribute to technical improvement.

Overall Table of Contents

1	Important Notes	1-1
1.1	Symbols	1-1
1.2	Safety Notes	1-1
1.3	Intended Use.....	1-1
1.4	Target Group.....	1-2
2	CAN	2-1
2.1	Data Objects	2-1
2.2	Identifier	2-1
2.3	PDO Communication	2-1
2.3.1	Indirect Process Data Communication	2-1
2.3.2	Data Exchange Sequence	2-2
2.3.3	Structure of Request and Response PDO	2-2
2.3.4	Structure of the Status Byte	2-3
2.3.5	Index Bytes	2-3
2.3.6	Subindex Byte.....	2-3
2.3.7	User Data Bytes of Request and Response PDO	2-4
2.3.8	Response Object With an Error	2-5
2.3.9	Tasks of the Communication Partner	2-5
2.3.9.1	Procedure for Communication Without Errors	2-5
2.3.9.2	Procedure for Communication With Errors	2-5
2.4	SDO Communication	2-6
2.4.1	Server-SDO	2-6
2.4.2	Client-SDO.....	2-6
2.5	CANopen in Operating Devices	2-6
2.5.1	NMT Capability	2-6
2.5.2	Object Dictionary	2-6
2.5.3	Object Description	2-8
2.5.3.1	Guard Time.....	2-8
2.5.3.2	Life Time Factor.....	2-8
2.5.3.3	Store Parameters.....	2-9
2.5.3.4	Restore Default Parameters	2-9
2.6	Programming	2-10
2.6.1	Protocol Parameters	2-10
2.6.1.1	Baud Rate.....	2-10
2.6.1.2	Maximum Waiting Time For Response.....	2-10
2.6.1.3	Delay until Connection Set-Up.....	2-10
2.6.1.4	Register Errors in Serial Message System	2-11
2.6.1.5	Use CANopen.....	2-11
2.6.1.6	Terminal Status on the Bus	2-12
2.6.1.7	Node Guarding for SDO Channels	2-12
2.6.1.8	Communication Relations.....	2-12
2.6.1.9	Terminal Module Number	2-13

Overall Table of Contents

2.6.1.10	Communication Over	2-13
2.6.1.11	Identifier Settings.....	2-14
2.6.2	Poll Area via PDO Channel.....	2-15
2.6.3	Poll Area via SDO Channel.....	2-15
2.6.4	Parallel Message System via PDO Channel.....	2-16
2.6.5	Parallel Message System via SDO Channel.....	2-16
2.6.6	Strings via the SDO Channel	2-16
2.6.7	Tables via the SDO Channel	2-16
2.6.8	Input Syntax	2-17
2.6.9	Physical Connection	2-18
2.6.9.1	Pin Assignment.....	2-18
2.6.9.2	Cable for CAN	2-18
2.7	Error Messages.....	2-19
2.8	Applications.....	2-21
2.8.1	Examples for Communication Relations	2-21
A	Index	A-1

1 Important Notes

1.1 Symbols

The symbols in this manual are used to draw your attention on notes and dangers.

**Danger**

This symbol is used to refer to instructions which, if ignored or not carefully followed could result in personal injury.

**Note**

This symbol indicates application tips or supplementary notes.

**Reference to source of information**

This symbol refers to detailed sources of information on the current topic.

1.2 Safety Notes

- Read this manual carefully before using the operating device. Keep this manual in a place where it is always accessible to all users.
- Proper transportation, handling and storage, placement and installation of this product are prerequisites for its subsequent flawless and safe operation.
- This user manual contains the most important information for the safe operation of the device.
- The user manual, in particular the safety notes, must be observed by all personnel working with the device.
- Observe the accident prevention rules and regulations that apply to the operating site.
- Installation and operation must only be carried out by qualified and trained personnel.

1.3 Intended Use

- The device is designed for use in the industry.
- The device is state-of-the-art and has been built to the latest standard safety requirements. However, dangerous situations or damage to the machine itself or other property can arise from the use of this device.
- The device fulfills the requirements of the EMC directives and harmonized European standards. Any modifications to the system can influence the EMC behavior.

1.4 Target Group

All configuration and programming work in connection with the automation system must be performed by trained personnel only (e.g. qualified electricians, electrical engineers).

The configuration and programming personnel must be familiar with the safety concepts of automation technology.

2 CAN

All operating devices with the CAN interface closely conform with the "CANopen Communication Profile" specification. The interface complies with the specifications of ISO-DIS 11898.

CANopen uses a subset of the CAL (CAN Applications Layer) communication services. The operating devices allow access to the data of devices on the CANopen bus by means of special PDOs or SDOs.

By defining up to 15 communication relations, the operating device can access the data of up to 15 different devices.

2.1 Data Objects

All CANopen data are located in an object dictionary with an index and a subindex. Data packets which can be exchanged on a CANopen network are referred to as communication objects. The communication objects are divided up into **PDOs** (process data objects) and **SDOs** (service data objects).

PDOs (process data objects):

- Maximum length = 8 bytes.
- Contain 1 to 64 data objects from the object dictionary. Their contents is defined by the device manufacturer.

SDOs (service data objects):

- Direct access to the object dictionary via index and subindex.
- Data of any size can be transmitted.

2.2 Identifier

Every communication object is identified by a unique identifier. Every PDO is a communication object to which an identifier is assigned. Two communication objects are always required for communications based on using SDOs. An identifier is assigned to each of the two communication objects.

2.3 PDO Communication

2.3.1 Indirect Process Data Communication

An operating device is generally capable of displaying or changing a data volume many times larger than that which can be exchanged using a PDO.

In addition, the PDOs predefined by the device manufacturer usually do not contain the data to be processed by the operating device. For this reason, the operating device does not access data objects directly but through special communication objects,

- a request PDO and
- a response PDO.

The request PDO and response PDO are used to establish a point-to-point connection on the data level between two devices.

2.3.2 Data Exchange Sequence

Operating device is the client

All services required for the operation of the operating device are initiated by the operating device itself, i. e. the operating device has client functionality.

Communication partner is the server

The communication partner merely responds to requests from the operating device, i. e. the communication partner has server functionality.

The operating device sends a request PDO and the communication partner returns a response PDO as a reply/acknowledgment.

If the communication partner receives a request object from the operating device, it interprets the first 4 bytes for data direction and accessed data object.

In the case of a write request, it stores the received data and returns a blank response PDO for acknowledgment.

In the case of a read request, it transfers the requested data to the operating device via a response PDO.

Request PDO and response PDO always exist in pairs. Each pair stands for one communication relation between the operating device and **one** station.

2.3.3 Structure of Request and Response PDO

Request and response PDO are each 8 bytes in size. The first four bytes of these objects contain the definition of the accessed data object from the object dictionary. The remaining 4 bytes contain the data.

Table 2-1 Structure of request and response PDO

Byte	Content
1	Status
2	Index Low
3	Index High
4	Subindex
5	User Data
6	
7	
8	

2.3.4 Structure of the Status Byte

The status byte within the request and response PDO has the following structure.

Table 2-2 Structure of the status byte in a request and response PDO

Bit	Designation	Value	Access	Description
0	Length	1, 2	Bit	Length of the Set/Delete Bit Mask
1		3, 4	Byte	Number in Bytes
2		2, 4	Word	Number in Bytes
		4	Double-Word	Number in Bytes
3	Toggle Bit			This bit toggles with each PDO. This indicates to the controller that a new PDO has been received.
4	Telegram Type	0 1		Request Telegram Response Telegram
5	Access	0	Bit	Bit Access in Write Mode (Set/Delete)
		1	Byte	Read or Write Byte
6		2 3	Word Double-Word	Read or Write Word Read or Write Double-word
7	Control Bit	0 1 0 1		Request PDO = Write Request PDO = Read Response PDO = No Error Response PDO = Error from the controller during telegram processing. The first data byte contains the error number.

2.3.5 Index Bytes

This value specifies the index from the communication partner's object dictionary. The index points to:

- one variable without subindex,
- the beginning of an array,
- the beginning of a record.

If you do not use CANopen, the numbering between communication partners can be declared here as desired.

2.3.6 Subindex Byte

This value specifies the subindex from the communication partner's object dictionary. The subindex points to one variable (basic CAN variable) of the size of the access.

If you do not use CANopen, the numbering between communication partners can be declared here as desired.

2.3.7 User Data Bytes of Request and Response PDO

The user data are classified by the type of access.

Table 2-3 User data, bit-oriented in a byte address

Byte	Content
5	AND Mask
6	
7	OR Mask
8	

Table 2-4 User data, bit-oriented in a word address

Byte	Content
5	AND Mask Low Byte
6	AND Mask High Byte
7	OR Mask Low Byte
8	OR Mask High Byte

The bit masks must be linked in the controller with the specified address using the OR or AND logical operations, where the AND mask deletes bits and the OR mask sets bits.

Table 2-5 User data, byte-oriented in a word address

Byte	Word	Content
5	1	Byte 1
6		Byte 2
7	2	Byte 3
8		Byte 4

Table 2-6 User data, double-word oriented in a word address

Byte	Double word	Content
5	1	Lowest Byte
6		Byte
7		Byte
8		Highest Byte

2.3.8 Response Object With an Error

If a communication error occurs, a response object with the following structure is returned.

Table 2-7 Structure of the response object containing an error

Byte	Content
5	Error Number
6	
7	
8	

2.3.9 Tasks of the Communication Partner

The TesiMod operating device's communication partner has a server functionality. As a server, the communication partner is responsible for interpreting incoming telegrams and for replying with a response object. This procedure must be performed within the waiting time specified in the protocol parameters.

A response object must be transferred in reply to every request object. In the case of an error, the communication partner simultaneously transmits an error number to the operating device.

2.3.9.1 Procedure for Communication Without Errors

1. The status byte of the request object is copied to the response object with the following change:

Bit 7	Control Bit	0	Error-free Action
Bit 4	Telegram Type	1	Response Object

2. Index and subindex are copied from the request object to the response object.
3. When the operating device writes data, the user data must be copied to the accessed address.
4. When the operating device reads data, the data must be copied from the accessed address to the address for the user data bytes.

2.3.9.2 Procedure for Communication With Errors

If the communication partner detects an error in the request object contents, this can be indicated to the operating device through an error message in the response object.

The error numbers can be freely defined. The operating device always displays the communication error "Communication Error 100". The subcode of the error message contains the error number.

1. The status byte of the request object is copied to the response object with the following change:

Bit 7	Control bit	1	Error detected
Bit 4	Telegram type	1	Response object
2. Index and subindex are copied from the request object to the response object.
3. The error number is entered into byte 5.

2.4 SDO Communication

SDO communications are point-to-point connections. One station is always the client while the other acts as a server.

With CANopen, a distinction is made between a client SDO and a server SDO.

2.4.1 Server-SDO

Each CANopen module has a server SDO channel. This channel allows external access to the object dictionary.

2.4.2 Client-SDO

In addition to a server SDO channel, the operating device also has a client SDO channel. This makes it possible to access the object dictionary using the server SDO channel of another device.

2.5 CANopen in Operating Devices

2.5.1 NMT Capability

The operating device is a "minimum capability device" in accordance with DS301 and thus provides minimum NMT functionality (PRE_OPERATIONAL and OPERATIONAL only).

2.5.2 Object Dictionary

Table 2-8 Object dictionary for CANopen

Index	Object	Name	Type	Attribute
0x1000	VAR	Device Type	Unsigned 32	const
0x1001	VAR	Error Register	Unsigned 8	ro
0x1004	ARRAY	Number of PDO Supported	Unsigned 32	ro
0x1008	VAR	Manufacturer Device Name	vis_string	const
0x1009	VAR	Manufacturer Hardware Version	vis_string	const
0x100A	VAR	Manufacturer Software Version	vis_string	const

Table 2-8 Object dictionary for CANopen

Index	Object	Name	Type	Attribute
0x100B	VAR	Node ID	Unsigned 32	ro
0x100C	VAR	Guard Time	Unsigned 16	rw
0x100D	VAR	Life Time Factor	Unsigned 32	rw
0x100E	VAR	Guard ID	Unsigned 32	ro
0x100F	VAR	Number of SDO Supported	Unsigned 32	ro
0x1010	VAR	Store Parameters	Unsigned 32	rw
0x1011	VAR	Restore Default Parameters	Unsigned 32	rw
0x1014	VAR	Emergency ID	Unsigned 32	ro
0x1200	RECORD	1st Server SDO	SDO Parameter	ro
0x1280	RECORD	1st Client SDO	SDO Parameter	ro
0x1281	RECORD	2nd Client SDO	SDO Parameter	ro
0x1282	RECORD	3rd Client SDO	SDO Parameter	ro
0x1283	RECORD	4th Client SDO	SDO Parameter	ro
0x1284	RECORD	5th Client SDO	SDO Parameter	ro
0x1285	RECORD	6th Client SDO	SDO Parameter	ro
0x1286	RECORD	7th Client SDO	SDO Parameter	ro
0x1287	RECORD	8th Client SDO	SDO Parameter	ro
0x1288	RECORD	9th Client SDO	SDO Parameter	ro
0x1289	RECORD	10th Client SDO	SDO Parameter	ro
0x128A	RECORD	11th Client SDO	SDO Parameter	ro
0x128B	RECORD	12th Client SDO	SDO Parameter	ro
0x128C	RECORD	13th Client SDO	SDO Parameter	ro
0x128D	RECORD	14th Client SDO	SDO Parameter	ro
0x128E	RECORD	15th Client SDO	SDO Parameter	ro

2.5.3 Object Description

Described below are only objects which differ from the standard definition.

2.5.3.1 Guard Time

Table 2-9 Guard time for CANopen

Designation	Value
Index	100Ch
Name	Guard Time
Object Code	VAR
Data Type	Unsigned 16
Object Class	Optional
Access	rw
Value Range	100 to 65535
Default Value	0

2.5.3.2 Life Time Factor

Table 2-10 Life time factor for CANopen

Designation	Value
Index	100Dh
Name	Life Time Factor
Object Code	VAR
Data Type	Unsigned 8
Object Class	Optional
Access	rw
Value Range	0 to 255
Default Value	0

2.5.3.3 Store Parameters

Table 2-11 Store parameters for CANopen

Designation	Value
Index	1010h
Name	Store Parameters
Object Code	ARRAY
Number of Elements	1
Data Type	Unsigned 16
Sub Index	4
Description	Save Manufacturer Parameters
Object Class	Optional
Access	rw
Value Range	65766173h
Default Value	no

2.5.3.4 Restore Default Parameters

Table 2-12 Store parameters for CANopen

Designation	Value
Index	1011h
Name	Restore Default Parameters
Object Code	ARRAY
Number of Elements	1
Data Type	Unsigned 32
Sub Index	4
Description	Restore Manufacturer Default
Object Class	Optional
Access	rw
Value Range	64616F6Ch
Default Value	no

2.6 Programming

2.6.1 Protocol Parameters

With the protocol parameters, you can adapt the communication of the controller used.

2.6.1.1 Baud Rate

This parameter specifies the communication rate.

Table 2-13 Baud rate

Configurable Values (Baud)	Default Value
10 kBit/s	
20 kBit/s	X
50 kBit/s	
100 kBit/s	
125 kBit/s	
250 kBit/s	
500 kBit/s	
1 MBit/s	

2.6.1.2 Maximum Waiting Time For Response

This parameter specifies how long the operating device waits for a response from the controller.

Table 2-14 Maximum waiting time for response

Configurable Values	Default Value
1 ms to 65535 ms	1000 ms

2.6.1.3 Delay until Connection Set-Up

This parameter specifies the waiting time after which the operating device starts the communication.

Table 2-15 Delay until connection set-up

Configurable Values	Default Value
1 s to 255 s	5 s

2.6.1.4 Register Errors in Serial Message System

If you activate this parameter, Node Guarding and NMT service related errors are recorded.

Table 2-16 Register errors in serial message system

Configurable Values	Default Value
ON	X
OFF	

Enter the following error numbers and error descriptions into the list of error messages.

Table 2-17 Error messages for the serial message system for CAN

Value	Text
General	
9020	Terminal is in PRE_OPERATIONAL State
9021	Terminal is in OPERATIONAL State
Only if Terminal Status = Master	
9012	Timeout Waiting for Response When Reading the Guarding Identifier
9013	Timeout Waiting for Response When Reading the Guard Time
9014	Timeout Waiting for Response When Reading the Life Time Factor
9015	Timeout During Guarding Telegram
9129 to 9255	Emergency telegram from SDO partners Error number = 9000 + 128 + node number of the emergency message

2.6.1.5 Use CANopen

If you activate this parameter, you indicate that you are using CANopen.

Table 2-18 Use CANopen

Configurable Values	Default Value
OFF	X
ON	

Activating this parameter allows you to choose the following parameters:

- Terminal Status on the Bus
- CANopen NMT service

2.6.1.6 Terminal Status on the Bus

Use this parameter to determine whether you want the operating device to function as the master or the slave. This applies to SDO accesses, NMT services and node guarding.

Table 2-19 Terminal status on the bus

Configurable Values	Default Value
Slave	X
Master	

If you choose the value „Slave“, the operating device - after initializing - expects a start/stop telegram from the master. In addition, it responds to NMT commands.

Table 2-20 NMT commands

NMT Command	Byte 0	Byte 1	Operating Device Reaction
Start node	1	0	OPERATIONAL
Stop node	2	0	PRE_OPERATIONAL
Enter pre-operational	128	0	PRE_OPERATIONAL
Reset node	129	0	Communication Error 60, Reboot, PRE-OPERATIONAL
Reset communication	130	0	Re-initialize CAN, PRE-OPERATIONAL

If you choose the value "Master", the parameter "Node guarding for SDO channels" is enabled and can then be selected. After the initializing phase, the operating device sends a global Start Node telegram to every station.

2.6.1.7 Node Guarding for SDO Channels

A limited node guarding is performed.

A slave, which is accessed by the operating device per SDO, expects a Start Node telegram. During operation, the operating device monitors all SDO channels to verify that the stations are in the "OPERATIONAL" state. It uses a Node Guarding telegram for this purpose. If a station is not in this state, a Start Node telegram is sent to this station to restore it to the "OPERATIONAL" state.

Table 2-21 Node guarding for SDO channels

Configurable Values	Default Value
OFF	X
ON	

2.6.1.8 Communication Relations

Define up to 15 independent communication relations in the table.

	Communication over	Identifier settings			Identifier		Default address type in bytes
		Mode	Connected with	Module number	Request identifier	Response identifier	
1	PDO1	Manual	Master	0	513	385	1
2	PDO1	CANopen	Master	10	394	522	2
3	PDO1	CANopen	Slave	20	532	404	4
4	PDO2	Manual	Master	0	385	513	1
5	PDO2	CANopen	Master	10	650	778	1
6	PDO2	CANopen	Slave	30	798	670	1
7	SDO	Manual	Master	0	385	513	1
8	SDO	CANopen	Master	10	1418	1546	1
9	PDO2	CANopen	Slave	40	808	680	1
10	PDO1	Manual	Master	0	385	513	1
11	PDO1	Manual	Master	0	385	513	1
12	PDO1	Manual	Master	0	385	513	1
13	PDO1	Manual	Master	0	385	513	1
14	PDO1	Manual	Master	0	385	513	1
15	PDO1	Manual	Master	0	385	513	1

Figure 2-1 Communication relations for CAN

2.6.1.9 Terminal Module Number

In CANopen mode, the assignment of the identifiers for the request and response objects are computed based on the module number.

Table 2-22 Terminal module number

Configurable Values	Default Value
0 to 127	1

2.6.1.10 Communication Over

This parameter specifies if the communication relation is to be handled using PDOs or SDOs.

Table 2-23 Communication over

Configurable Values	Default Value
PDO1	X
PDO2	
SDO	

2.6.1.11 Identifier Settings

In CANopen mode, the identifiers are assigned automatically. If this mode is not used, you need to assign values for the request and response identifier manually.

Mode:

This parameter indicates whether you want to enter the identifier values manually or want the values to be computed automatically in accordance with the CANopen default values.

Table 2-24 Mode

Configurable Values	Default Value
Manual	X
CANopen	

Connected with:

This parameter specifies whether the operating device communicates with a master or with a slave. Based on this information, the values for the request and response identifiers are swapped. This parameter is only available if you select CANopen for the **Mode** option.

Table 2-25 Connected with

Configurable Values	Default Value
Master	X
Slave	

Module number:

This parameter specifies the slave's module number for a communication relation in the CANopen mode. To be able to change the module number, you must select the following settings

- **Mode** = CANopen and
- **Connected with** = Slave.

Table 2-26 Module number

Configurable Values	Default Value
0 to 127	0

Identifier:

In the CANopen mode, the identifier values for request and response objects are computed automatically. In Manual mode, you need to enter the values manually.

Table 2-27 Identifier

Configurable Values	Default Value
0 to 65535	385 (Request), 513 (Response)

Default address type in bytes:

This parameter specifies the address type of the data block of the communication partner. This parameter is relevant for PDO communication relations only.

Table 2-28 Default address type in bytes

Configurable Values	Default Value
1 (Byte)	X
2 (Word)	
3 (Double-word)	

2.6.2 Poll Area via PDO Channel

For the poll area, specify whether you want to use one or three variables for this purpose.

Restrictions regarding the poll area with 1 variable:

- The variables must be byte-oriented or word-oriented.
- The area must be contiguous.
- The controller must be able to access this area in bit-mode.
- The operating device must be able to access this area in word-mode.

Restrictions regarding the poll area with 3 variables:

- The variables must be byte-oriented or word-oriented.
- The area for the single variable LED Control must be contiguous.
- The controller must be able to access these areas in bit-mode.
- The operating device must be able to access these areas in word-mode.

Table 2-29 Word-oriented polling area for CAN

Word Address	Reference	High Byte	Low Byte
Word Address + 0	Index 100 Subindex 0	WCB	Reserved
Word Address + 1	Index 100 Subindex 2	Message Channel High Byte	Message Channel Low Byte
Word Address + 2	Index 100 Subindex 4	LEDs 1 to 4	LEDs 5 to 8
Word Address + 3	Index 100 Subindex 6	LEDs 9 to 12	LEDs 13 to 16
Word Address + 4	Index 100 Subindex 8	LEDs 17 to 20	LEDs 21 to 24
Word Address + 5	Index 100 Subindex 10	LEDs 25 to 28	LEDs 29 to 32

2.6.3 Poll Area via SDO Channel

You can define the poll area via SDO channel for one single variable only.

The data for the poll area must be created as a "segmented object" on the server for this object. The object for deleting the message number in the serial message channel must be created using the following subindex:

- Poll area is byte-oriented = subindex 2
- Poll area is word-oriented = subindex 1

2.6.4 Parallel Message System via PDO Channel

The same principles that apply to the polling area apply to the data area for the parallel message system.

- The data area must be byte-oriented or word-oriented.
- A byte address can be accessed in byte-word or word-mode.
- A word address can be accessed in word-mode only.

2.6.5 Parallel Message System via SDO Channel

The data for the parallel message system must be created as a "segmented object" on the server for this object.

2.6.6 Strings via the SDO Channel

Strings must be created as a "segmented object". The size of strings can not exceed 48 bytes.

2.6.7 Tables via the SDO Channel

Data objects to be displayed as tables must be created on the server as an "array". The individual elements of an array are accessed via the subindex of the object.

Each subindex points to one element in the table. The table can contain 256 elements as a byte, word or double-word.

2.6.8 Input Syntax

The following image illustrates the structure of the input syntax for variables in the programming software.

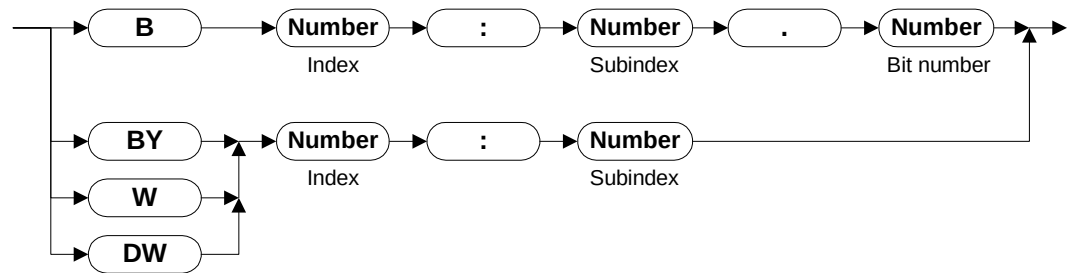


Figure 2-2 Syntax diagram

2.6.9 Physical Connection

The operating devices to be connected to the CAN bus are equipped with interfaces complying with the CiA Draft Standard 102. for operating devices with two interfaces all signal lines are jumpered from connector to connector.

2.6.9.1 Pin Assignment

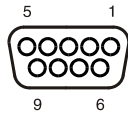


Figure 2-3 9 pin D-SUB female connector strip

Connector in the terminal: 9 pin D-SUB female connector strip.

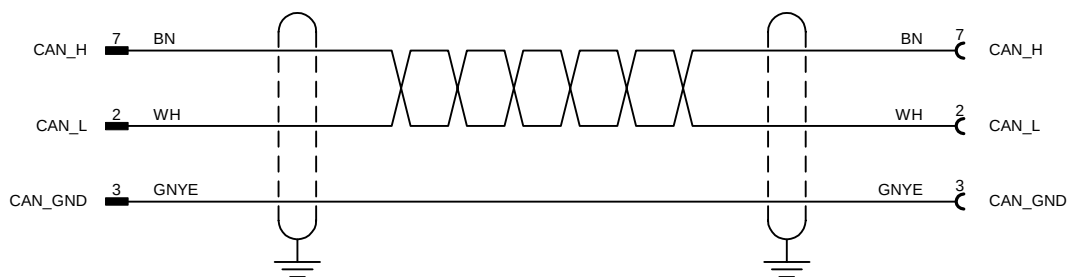
Table 2-30 Pin assignment CAN

Pin	Designation	Function
1	nc	Not Connected
2	CAN_L	CAN_L Bus Line (Dominant LOW)
3	CAN_GND	CAN Ground
4	nc	Not Connected
5	nc	Not Connected
6	CAN_GND	CAN Ground
7	CAN_H	CAN_H Bus Line (Dominant HIGH)
8	nc	Not Connected
9	nc	Not Connected

2.6.9.2 Cable for CAN

Operating Device

Next CAN Bus Participant



D-SUB
Male Connector
9 Pin

D-SUB
Female Connector
9 Pin

2.7 Error Messages

Error messages are displayed on the operating device along with a code and sub-code. Error messages are composed as follows:

Communication Error

Code XXXXX

Subcode XXXXX

Retries XXXXX

Table 2-31 CAN error messages

Code	Subcode	Error Type	Possible Cause
1		Slave is currently not ready	
2		Packets out of sequence	
3		Protocol framing error	
4		Timeout	
5		CRC error	
6		Parity error	
7		Send process aborted	
8		Receive process aborted	
9		Buffer too small for cyclic data	
10		No cyclic data defined	
12		Cyclic data already defined	
15		The selected protocol is not supported	
16		Receive buffer overrun	
40		Undefined system variable	
50		Error from CAN controller Error number is stated in subcode	
	1	Stuff error	
	2	Format error	
	3	Operating device has no connection to bus	No stations are connected to the bus.
	4, 5	Bus error	Bus line has short-circuit to logical 0 or 1
	6	CRC error	
51		No response from communication partner	No partner for request object or response object. No recipient for transferred identifier.
	xx	Identifier, where the error occurred.	

Table 2-31 CAN error messages

Code	Subcode	Error Type	Possible Cause
53		Wrong response object	Message without response bit (PDO)
	xx	Identifier, where the error occurred	
54		Wrong response object	Response without request
	xx	Identifier, where the error occurred	
55		Operating device can not find CAN hardware.	CAN hardware in operating device is defective
	0		
60		NMT 0 message from master with module reset (129)	
	0		
100		Error from communication partner Error number is stated in subcode	
	xx	Error number from communication partner - freely definable	
System errors			
1283		Parameter inconsistent. Toggle bit does not change	
1284		Waiting time has elapsed	
1537		Write access to read-only object	
1538		Object does not exist	
1542		Error during access due to a hardware error	
1543		Data type conflict	
1545		Subindex does not exist	
2048		Other error	

2.8 Applications

2.8.1 Examples for Communication Relations

With TSWin, you can define up to 15 communication relations. Carry out the following steps to open the table for the communication relations:

1. In the main window of TSWin, select the **Controllers** tab.
2. Select the **Communication parameters** option from the left list box.
3. Click the **Edit** button.

The **Protocol parameters CAN** window appears.

4. Click the **Communication relations** button.

The **CAN Communication relations** window opens.

	Commu- nication over	Identifier settings					Default address type in bytes
		Rules			Identifier		
		Mode	Connected with	Module number	Request identifier	Response identifier	
1	PD01	Manual	Master	0	100	200	1
2	PD01	CANopen	Master	5	389	517	1
3	PD01	CANopen	Slave	10	522	394	1
4	PD01	CANopen	Slave	20	532	404	1
5	PD01	CANopen	Master	5	389	517	2
6	PD01	CANopen	Master	5	389	517	4
7	PD01	Manual	Master	0	385	513	1
8	PD01	Manual	Master	0	1418	1546	1
9	PD01	Manual	Master	0	808	680	1
10	PD01	Manual	Master	0	385	513	1
11	PD01	Manual	Master	0	385	513	1
12	PD01	Manual	Master	0	385	513	1
13	PD01	Manual	Master	0	385	513	1
14	PD01	Manual	Master	0	385	513	1
15	PD01	Manual	Master	0	385	513	1

Figure 2-4 Communication relations for CAN

Below see an example of a variable list with sample variables for these communication relations.

Table 2-32 Variable list for CAN

No.	Variable name	Address	Comm. Relation.:
0	Var1	B 0:1.3	1
1	Var2	B 100:10.12	5
2	Var3	B 100:10.12	1 / 2
3	Var4	BY 1:35	1
4	Var5	BY 10:78	1
5	Var6	W 11:24	1
6	Var7	W 100:102	5
7	Var8	W 130:12	1 / 2
8	Var9	DW 100:102	5
9	Var10	DW 130:20	1 / 2
10	Var11	DW 50:25	6
11	Var12	DW 200:20	1 / 4

Explanation of the addresses

The following applies to the communication relation 1:

- Index 0 to 99 is byte-oriented
- Index 100 to 199 is word-oriented
- Index 200 to 300 is double-word oriented.

Table 2-33 Explanation of the addresses

Variable Name	Access	Index	Sub-index	Bit Number	Communication Relation	Address Type
Var1	Bit	0	1	3	1	
Bit access: Bit 3 of index 0, subindex 1 of communication relation 1 If no address type is entered, the default value in the communication relation definition table applies. In this case, no specified address type means = byte address.						
Var3	Bit	100	10	12	1	2
Bit access: Bit 12 of index 100, subindex 10 of communication relation 1 Index 100 in the communication relation 1 is a word address. However, the default setting is "Byte address". Therefore, a "2" must be entered here for the address type.						
Var2	Bit	100	10	12	5	
Bit access: Bit 12 of index 100, subindex 10 of communication relation 5						
Var5	Byte	10	78		1	
Byte access: Index 10, subindex 78, communication relation 1						
Var8	Word	130	12		1	2
Word access: Index 130, subindex 12, communication relation 1 Index 100 to 199 in communication relation 1 is a word address. However, the default setting is "Byte address". Therefore, a "2" must be entered here for the address type.						

Table 2-33 Explanation of the addresses

Variable Name	Access	Index	Sub-index	Bit Number	Communication Relation	Address Type
Var12	Double word	200	20	3	1	4
Double-word access: Index 200, subindex 20, communication relation 1 Index 200 to 300 in communication relation 1 is a double-word address. However, the default setting is "Byte address". Therefore, a "4" must be entered for the address type.						
Var11	Double word	50	25		6	
Double-word access: Index 50, subindex 25, communication relation 6						

A Index

C

Cable

CAN 2-18

CAN 2-1

I

Important notes 1-1

Intended use 1-1

P

Pin assignment

CAN 2-18

Protocol parameters

CAN 2-10

S

Safety notes 1-1

Symbols 1-1

T

Target group 1-2



Sütron electronic GmbH
Kurze Straße 29
D-70794 Filderstadt
Phone: 0049 711 / 77098-0
Fax: 0049 711 / 77098-60
E-mail: doku@suetron.de
Internet: www.suetron.com

